Performance of Transport Protocols over a Multicasting-based Architecture for Internet Host Mobility

Jayanth P. Mysore Vaduvur Bharghavan
Coordinated Sciences Laboratory
University of Illinois at Urbana-Champaign
{babu, bharghav}@crhc.uiuc.edu

ABSTRACT: **IP multicasting provides a mechanism for location independent addressing and packet delivery to a group of hosts that belong to a multicast group. It also provides efficient mechanisms for hosts to join and leave multicast groups. The problem of supporting macro and micro level host mobility in the Internet involves similar issues of location independent addressing, address translation, packet forwarding and location management of mobile hosts. We have developed a new approach called MSM-IP for supporting host mobility using IP multicasting as the sole mechanism for addressing and routing packets to mobile hosts.**

In this paper, we present the performance of UDP and TCP over MSM-IP, and compare it with their performance over Mobile-IP. Our analysis shows that our approach is very effective in supporting seamless mobility for both handoffs and interface changes within and across networks. The performance of both UDP and TCP are better when used with MSM-IP than with Mobile-IP.

We briefly explain our approach, and present performance results, and explain the results.

1. OVERVIEW

In the Internet, a host is identified by a unicast IP address. Since IP uses this address for routing messages to the host, it must be distinct and location dependent. TCP exploits the uniqueness of a host's IP address to build connection identifiers for connections that originate or terminate at the host using a 4-tuple: <src addr, src port, dst addr, dst port>. This connection identifier is built at the start of the connection and is expected to remain invariant for the duration of the connection. While the above scheme works well in the domain of static hosts and connections, the requirements of IP and TCP prove to be conflicting when a host is mobile. While IP requires the host address to change in order to reflect it's current location, TCP requires the host address to remain unchanged for ongoing connections. The former is needed for correct datagram delivery, the latter for correct connection management. In order to resolve the dual nature of addressing, most current mobility support protocols provide a two layer addressing scheme for a mobile host - a *home address* which is a static identifier that remains unchanged even as a host moves, and a *care-of address* which is the dynamic identifier that reflects the current point of attachment. Mobility support protocols perform the mapping between the two addresses transparently to static end hosts, and possibly even the mobile hosts [3].

In a related work [16], we have explored an alternative mechanism to support Internet host mobility by exploiting the commonality of the problems faced by multicasting and mobility. Since IP-multicasting requires the support of multiple destinations in a single shared communication channel, the channel address is location independent. Since end hosts may dynamically join and leave the multicast group, IP-multicasting must address the problems similar to 'handoff' (i.e. joins and leaves). In the IP-multicasting architecture, each multicast group is identified by an IP address. Multicast datagrams are directed to this address and are snooped by multicast routers, which then 'tunnel' these packets to the end destinations. The key elements of IP multicasting that are relevant to the support of mobility are the following:

- The multicasting infrastructure provides a mechanism for location independent addressing and routing.
- The decision to join and leave a multicast channel lies entirely with the receiver, and typically terminates at the lowest levels of the routing tree hierarchy.
- The unicast IP address for end hosts may be used for network management but not for routing.

We can therefore view the multicasting architecture as fundamentally providing a mechanism for location independent addressing and routing. While multicasting services use this mechanism to identify multicast channels and route messages to those channels, we propose a new architecture called MSM-IP (Mobility Support using Multicasting in IP) in [16] to use the service to identify mobile end-points and route messages to them. The key idea in MSM-IP is to *assign multicast addresses to identify mobile hosts*. Whenever a mobile host moves into a new network, it joins the multicast group corresponding to it's own address. Therefore all communications to mobile hosts will be handled by multicast routers, which implement a multicast routing protocol such as DVMRP [24]. Note, that packets from a mobile host to a static host follow traditional unicast routing mechanisms, while packets from a static host to a mobile host are multicast to it. As we discuss in the related work, this architecture raises several interesting issues relating to location management, security and scalability. Additionally, though no changes are conceptually required for a multicasting architecture to support mobility, in practice, small changes are involved in several protocols such as TCP, IP, ARP, IGMP, etc.

We have implemented MSM-IP in our testbed, and studied its performance. The purpose of this paper is to provide a detailed understanding of the performance benefits of an MSM-IP approach over the traditional Mobile IP approach. We show an improvement in performance of both TCP and UDP with MSM-IP over Mobile IP. It is not the purpose of this paper to claim that MSM-IP is a currently viable solution for widespread deployment over the Internet. Indeed, several problems need to be resolved in the multicasting domain including security, location management/discovery, and efficient sparse routing before a general purpose multicasting

architecture for supporting Internet host mobility can be achieved. However, this paper does point out that the research issues in multicasting subsume the research issues in mobility, and also that using multicasting as the sole mobility support mechanism is very attractive in terms of performance. As part of on-going research we are proposing modifications to existing multicast routing protocols, so that the multicasting service matures into a robust integrated solution for all classes of location independent data delivery services in the Internet.

The rest of the paper is organized as follows. Section 2 describes the MSM-IP architecture and implementation. Section 3 presents the performance of UDP and TCP over MSM-IP. Section 4 compares our work with related work. Section 5 concludes the paper.

2. MSM-IP: Architecture and Implementation

MSM-IP uses the existing multicasting architecture for mobility support. In this section, we explain how the IP-multicasting architecture is used as a mobility support architecture.

The IP-multicasting architecture overlays a virtual interconnection of tunnels over the underlying internet. Tunnels are virtual connections connecting multicast routers. Multicast datagrams are routed via these tunnels along a distribution tree. The distribution tree is rooted at the source, and might change in topology during the course of an on-going multicast transaction session. This change in topology is due to new hosts joining the tree and previously participating hosts leaving it. Any host on the internet can send messages to the tree, provided it is connected to the virtual network of tunnels via a multicast router. The topology of the multicast tree is determined by the distribution of participating hosts on the tunneled network of multicast routers. Whenever a host wishes to listen to data on a channel, it builds a connection to it. Likewise when a host no longer wishes to receive data on a channel, it tears down the connection it previously built. Hosts register membership information with multicast routers serving their local network, using the IGMP protocol. The routers implement a multicast routing protocol (e.g. DVMRP [24], MOSPF [15], PIM[8], etc.) to set up multicast distribution trees.

In the MSM-IP architecture, static hosts are assigned standard unicast IP addresses. Mobile hosts are assigned unique *multicast IP addresses*. Thus, packets from a mobile host to a static host will be routed by standard unicast mechanisms while packets from a static host to a mobile host will be routed by the multicasting infrastructure. When a mobile host moves into a new network, it sends an IGMP registration message to the local multicast router, registering membership with the multicast group identified by it's own address. The multicast router now initiates a join procedure to join the multicast distribution tree. In the common case of mobility, we expect the current multicast router (the current network of a mobile host) to be near the previous multicast router (the previous network) that was servicing the mobile host. Therefore, we anticipate that the join will terminate at the lowest level of the distribution tree hierarchy. Once a host moves out of a network, it stops sending membership refresh messages to the old multicast router. This router eventually times the membership out, and initiates a prune. Like joins, prunes will terminate at the lowest levels of the distribution tree. Therefore, handoff using the IP multicasting architecture will be very efficient. The use of multicasting to reduce handoff dropping has been proposed elsewhere in literature [21]; however, to our knowledge, MSM-IP is the first proposal to use multicasting as the *sole* mechanism for supporting host mobility.

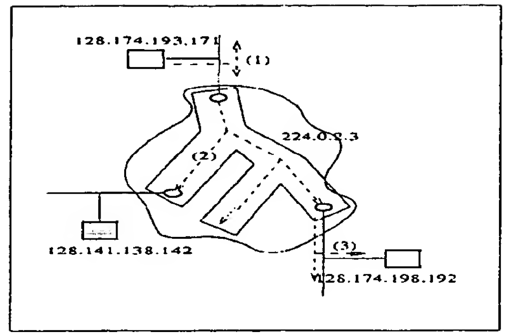

Fig. 1. This figure shows the multicasting architecture. In Step (1), a host transmits a multicast packet. In Step (2), the multicast router tunnels the multicast packet along the multicast distribution tree. In Step (3), each destination multicast router decapsulates the tunneled packet and multicasts it locally in its subnet. In MSM-IP, we use the same architecture. We assign mobile hosts multicast addresses, and make them join multicast groups corresponding to their own addresses when they move into a new network. This ensures that the local mrouted joins the multicast distribution tree.

2.1. Design and Implementation issues

Conceptually, the IP multicasting architecture can support host mobility without any changes to the existing infrastructure by simply assigning a unique multicast IP address to a mobile host. However, practically there are some important issues to be resolved before MSM-IP can be deployed for widespread use. Some of these are changes that need to be implemented in the protocol stack at the end-hosts and multicast routers, while others are enhancements that are needed to the MSM-IP architecture.

The protocol level issues that need to be addressed are in the implementation of the ARP, TCP, IGMP and ICMP. In the standard Linux kernel (version 2.0.30), we found that an ARP response is discarded by the portable, if it is assigned a multicast address. We currently set the ARP cache of the mobile host manually to by-pass the problem. There is currently no support for TCP over IP-multicasting. It can be exceedingly complex to provide TCP support when there are several hosts participating in a multicast transaction. However, in MSM-IP, we are guaranteed to have only one sender and one receiver. Providing TCP support over IP-multicasting thus reduces to letting TCP accept incoming connections from and permit outgoing connections to Class D addresses. A more detailed description of the modifications required in TCP is provided in section 3.1. When a host sends an IGMP registration message, it sends the membership message to the address of the multicast group it wishes to join. This destination address gets mapped to a multicast hardware address. Multicast routers pick up all messages addressed to multicast hardware addresses. We observed that when a host is assigned a multicast address, and tries to join the group identified by the same address, it maps the multicast destination address to it's own hardware address. Therefore, the local mrouted does not care to pick up the registration

message. We have changed the function in the portable's kernel that does the address mapping, so that it maps the destinations to the multicast hardware address. One other problem with IGMP is that the IGMP module running on the multicast router looks at the source address of a registration message to determine the interface on which the registration request arrived. When the source has a multicast address, the mrouted is unable to locate the sender's network. ICMP messages cannot be sent to a multicast destination, and hence if an ICMP message requires to be sent back to a mobile host under MSM-IP, there will be a problem. For the last two issues, the basic cause of the problem is that the two services require location dependent addresses for hosts. Therefore we propose using DHCP to allocate location specific addresses to mobile hosts, purely for network management functions. This provides a natural solution to routing problems - use location independent addresses to support mobility and location dependent addresses to support network management.

3. TESTS PERFORMED AND PERFORMANCE ANANLYSIS

The MSM-IP architecture has been implemented and operational in our laboratory testbed environment for about a year now. In our testbed, the backbone environment consists of two subnets, each of which is a 10Mbps switched Ethernet. All the network switches are 200 MHz Gateway Pentium Pros. The correspondent host used for testing is a Gateway Pentium 133 MHz, and all the mobile hosts used for testing are P6-120 TI laptops. All the hosts in the backbone run Linux 2.0.30. The correspondent host and the mobile host run a modified version of Linux-2.0.30. The testbed is shown in Figure 2. The modified kernel allows TCP to open connections to and accept connections from Class D addresses. Each subnet has a mrouted 3.81 multicast router. The wireless network is a 2.4GHz WavelanII, with one access point in each subnet. All mobile hosts have network access using both Ethernet and Wavelan cards. Each mobile host runs a registration module which registers with the current mrouted, and sends periodic refresh messages to keep alive the registrations. The applications we run in the testbed include WWW Browsers, ftp, Mpeg video over the network, editors, and file system access as well as special workloads generated for the purpose of quantifying the performance of our approach.

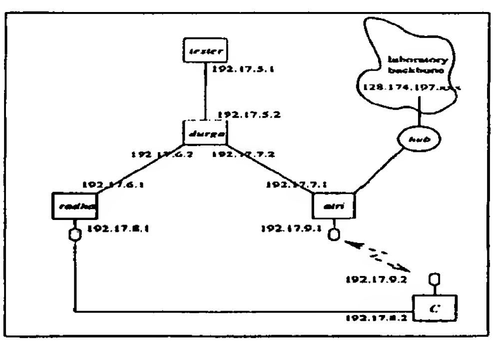

Fig. 2. The testbed configuration. Note, that *durga* can be configured to introduce a 100ms delay in packets without throttling down the bandwidth of its outgoing links in order to simulate network delay.

We have tested our approach with variations in offered workload, applications, types of handoff, and parameters of measurement. In this paper, we present a subset of these measurements. In terms of offered load, we used offered throughputs of 10Kbps, 100Kbps, 512 Kbps, 1Mbps, for UDP traffic, and TCP at peak traffic. In terms of packet sizes, we used 512 bytes and 1 KB packets for the user generated workloads, and also used the application programs (WWW browser, ftp, Mpeg) unchanged. In terms of offered traffic patterns, we used constant interpacket delay, packet bursts, and exponential interpacket delay distributions. In terms of handoffs, we performed two types of handoffs: (a) *hot* switches, in which the mobile host registered with the new mrouted before handing off, and (b) *cold* switches, in which the mobile host handed off before registering with the new mrouted. Hot switches are a close simulation of what happens in the common case of mobility between overlapping cells. While the heuristics for predicting the next cell accurately are beyond the scope of this paper, they are explored in a related work [14]. In each case, we experimented with handing off between cells, as well as changing device interfaces. Our interface changes experiments included Ethernet to Wavelan, Wavelan to Ethernet, and Ethernet to Ethernet. In all the above cases, we experimented with changing interfaces and handing off within the same subnet as well as across subnets. The parameters we measured included packet loss upon cold switches, packet duplicates upon hot switches, and the variation of TCP sequence numbers with time.

In this section, we briefly present the results we obtained using UDP as the transport protocol. A more detailed discussion of the results for UDP can be found in [16]. Next, we present the plots of sequence number against time during hot and cold switches when using TCP as the transport protocol.

3.1. *Software*

We use the Linux-2.0.30 kernel on all the machines in the testbed. A few changes were required in the networking software on the portable and the correspondent host. The changes that were required were in :

- TCP : 2 changes were made - 1 in the portable and 1 in the correspondent host.
- IP : 1 change was made on the portable.

Changes in TCP:

In the function tcp_connect, the host checks the nature of the remote address, before permitting the connect to go through. If the destination is a multicast address, the connect request is refused. We removed this check in TCP on the correspondent host, as it will have to open connections with a mobile host having a Class D address. In the function tcp_rcv, TCP checks the packet type of an incoming packet before processing it. TCP processes only packets of type PACKET_HOST. The packet type is set by the driver software based on the hardware address of the destination. In the case of a packet coming to a multicast destination, the hardware address will be a multicast hardware address. Hence, the packet type is set as PACKET_MULTICAST, which TCP discards. So, we modified TCP so that it processes both packets of type PACKET_HOST and PACKET_MULTICAST.

Changes in IP:

When a host sends out an IP datagram, the function ip_chk_addr is called to determine the nature of the destination address. Depending on the value returned, the destination address is mapped to a unicast Ethernet address (by invoking ARP, for example), or to a multicast Ethernet address. Now, ip_chk_addr *first* checks if the given address corresponds to one of the *machine's own addresses*, and then checks if the address corresponds to a multicast group. Consider the case when a portable is assigned a multicast address. When such a machine sends out an IGMP registration message to the multicast group given by it's own address, the ip_chk_addr function returns that the destination corresponds to one of the machine's own addresses. Consequently, the address gets mapped to one of the portables own interfaces. Consequently, the messages are sent to the portable's own interface address, and do not get picked up by the local multicast router. We modified ip_chk_addr on the portable so that it *first* checks if the destination is a *multicast address*, before checking if it corresponds to one of the machine's own addresses.

3.2. *UDP performance*

UDP runs unmodified over MSM-IP. Causing either interface change or handoffs across two different subnets involves more overhead as opposed to moving between points on the same subnet. This is because, a change in the network of connection requires the new multicast router to join the multicast tree, before the host can receive data. The MSM-IP performance is good even for interface switching at relatively good throughputs for wireless networks. The maximum packet loss we observed was 1, and we recorded a maximum of three duplicates. The performance was studied at throughputs of 100 Kbps and 400 Kbps, under constant and Poisson traffic distributions. The two networks shared a common gateway two hops away. The results of our performance measurements for switching across different networks is presented in Table 1. Note, that the results presented represent not only a handoff across neighboring networks, but also an interface change. A more detailed analysis of the performance of UDP over MSM-IP is provided in [16].

Mode	Flow	T'put	Model	Loss	Dup
Cold	Down	100	Constant	0.25	0
Cold	Up	100	Constant	1	0
Cold	Down	400	Constant	1	0
Cold	Up	400	Constant	1	0
Hot	Down	100	Constant	0	0.75
Hot	Up	100	Constant	1	0
Hot	Down	400	Constant	0	2.5
Hot	Up	400	Constant	1	0
Cold	Down	100	Poisson	0.75	0
Cold	Up	100	Poisson	1	0
Cold	Down	400	Poisson	1	0
Cold	Up	400	Poisson	1	0
Hot	Down	100	Poisson	0	0.75
Hot	Up	100	Poisson	0.75	0
Hot	Down	400	Poisson	0.33	1.75
Hot	Up	400	Poisson	1	0

TABLE I
PERFORMANCE OF MSM-IP WITH HANDOFF ACROSS TWO DIFFERENT SUBNETWORKS. 'MODE', 'FLOW', 'T'PUT' AND 'MODEL' INDICATE THE SWITCHING MODE, DIRECTION OF THE FLOW, OFFERED UDP THROUGHPUT (IN KBPS) AND TRAFFIC MODEL RESPECTIVELY. 'LOSS' AND 'DUP' INDICATE THE AVERAGE PACKET LOSS AND DUPLICATES THAT WERE MEASURED BY THE RECEIVING APPLICATION.

3.3. *TCP performance*

To study the performance of TCP over MSM-IP, we studied the distribution of the sequence numbers of the packets received against time, during a switch. All the switches were across different sub-nets, and included an interface change. We studied both switches across homogeneous networks(Ethernet to Ethernet) and heterogeneous networks(Ethernet to Wavelan). The results we obtained show that a hot switch under MSM-IP does *not* alter the performance of TCP in any noticeable manner. In the case of a cold switch, we notice a very brief period when the sender gets into slow start immediately after a switch. However, this lasts for only about 2-3 ms, after which the sender's window synchronizes with the mobile host's. The results are explained in greater detail in this section.

First, we present the results obtained for a hot switch across two Ethernet interfaces. The interfaces were connected to two different sub-nets. We observed that the performance of TCP is almost unaffected by a hot switch across Ethernets. The plot of sequence numbers against time is presented in figure 2. The switch over was initiated about 2.2 seconds after the start of data transfer. We observe that for a period of about a second after the switch over, there is a decrease in the rate at which the sequence numbers increase with time. When we examined the tcpdump outputs for the two interfaces, we found that it is exactly the time during which both the interfaces were up. During this period the portable had to do extra-processing, and therefore the rate at which it could process incoming packets was reduced. Note that the portable is a 120 MHz Pentium machine with 24 MB of RAM, while the sender is a 133 MHz Pentium with 64 MB of RAM. Consequently, the size of the advertised window of the mobile host reduces during this period, resulting in the sender sending packets in smaller windows. As a result, we observe a smaller rate of increase in the sequence numbers. At the end of a second, the original interface goes down, and we immediately see a restoration in the rate of increase in the sequence numbers. To illustrate the decrease in the slope of the sequence number vs time plot, we studied the rate of increase of sequence numbers, when we had one and two interfaces up. The graphs obtained are presented in Figure 3.

Next, we present the results obtained when we performed a cold switch across two Ethernet interfaces. As before, the two interfaces were connected to different sub-nets. The switch was performed about 2.8 seconds after the initiation of data transfer. In the case of a cold switch, the current interface is disabled before enabling the new interface. Immediately after initiating the switch operation the mobile host is therefore disconnected. Consequently, the ACK sent by the portable does not get across the network. So, after 200ms, which is one round-trip time, we see that the sender retransmits the last sent segment. After disabling the previous interface, the new interface is enabled to receive packets. However, it takes about 300 more milliseconds before the routing tables for the new interface are set up. Hence, the mobile host will be unable to send any ACKs for that much longer. After waiting for 400ms, which is two round trip times,

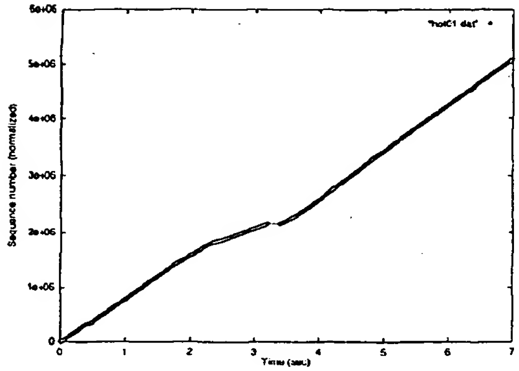

Fig. 3. TCP performance under MSM-IP, during a hot switch across Ethernets

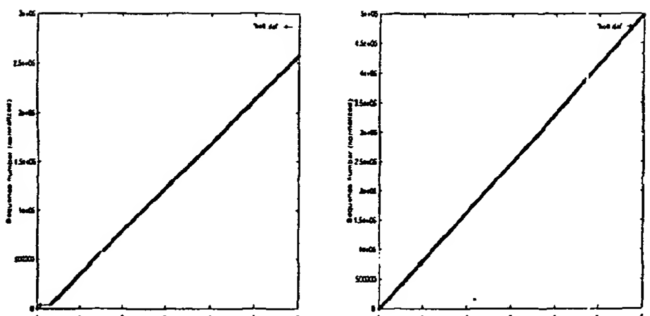

Fig. 4. TCP performance with one(fig. on left) and two(right) interfaces enabled. Note that the slope of the plot is almost doubled when only one interface is enabled, when compared to the slope when both interfaces are enabled.

the sender's congestion window falls to one, and the sender enforces congestion control by going into slow-start. This is seen in the graph as a brief exponential rise. However, by this time the portable is fully ready to send ACKs, and the portable acknowledges incoming packets. Very soon the portable catches up, and the sender gets out of the slow-start phase.

Finally, we present the results for the case when a mobile host moves from an Ethernet to a Wavelan cell. Again, the Ethernet and the Wavelan interfaces were connected to different sub-nets. The switch was effected about 1.7 seconds after the initiation of data transfer. In the initial period after switching into Wavelan, we observe some retransmissions. From the figure, it is clear that one fast retransmit a little after 2 seconds causes the congestion window to halve and introduces congestion avoidance. However, we are unable to explain the reception of the two packets at around 2 seconds with sequence numbers < 110000. With the exception of these two stray retransmissions, the handoff takes place in a smooth manner with the initiation of fast recovery and congestion avoidance.

3.4. Comparison with Mobile IP

In this section we compare the performance of TCP over Mobile-IP against it's performance over MSM-IP. For Mobile IP, the correspondent host was co-located with the home in order to eliminate the effect of triangular routing, and the foreign agent was co-located with the mobile host. Additionally, we implemented some performance enhancing mechanisms to eliminate registration. In our implementation, a mobile host registers all the IP addresses it is likely to get assigned during the course of a connection, with the correspondent host. When a mobile host moves into a new network, it does not have to register with the home agent as in Mobile IP. The correspondent host detects the change of address, and ensures it is one of the registered addresses.

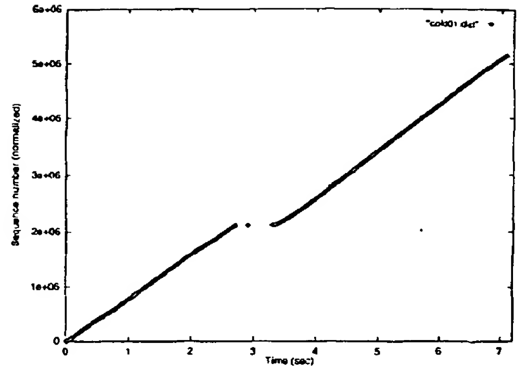

Fig. 5. TCP performance under MSM-IP, during a cold switch across Ethernets

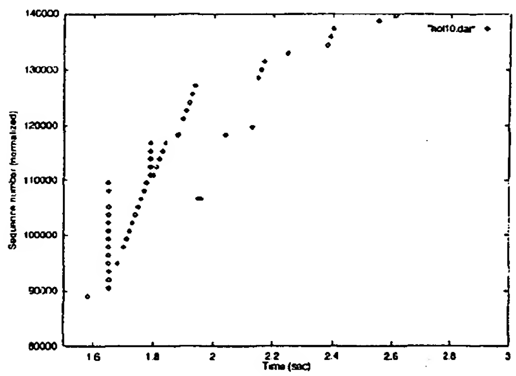

Fig. 6. TCP performance under MSM-IP, during a hot switch form an Ethernet to a Wavelan network. Note that the graph shown indicates the variation in sequnce numbers around the time of switching. All the other graphs presented a more overall picture

First we present the performance of TCP over Mobile-IP, on a network with a negligible delay bandwidth product. In such a network, the number of packets lost during a switch will be very few. Consequently, TCP's congestion control mechanisms do not get activated. Figure 6 shows the variation in the sequence numbers against time. The switch was performed approximately 7 seconds after the initiation of data transfer. TCP recovers in about 200ms.

However, when we increase the delay of the network significantly, a large number of packets can potentially get lost during a switch over. This is because, until the ACKs sent out of the new interface reach the correspondent host, the correspondent does not realize a change in the mobile host's address. Consequently, it keeps sending data to the old interface, which is no longer enabled. All these data packets get lost. TCP interprets this loss as con-

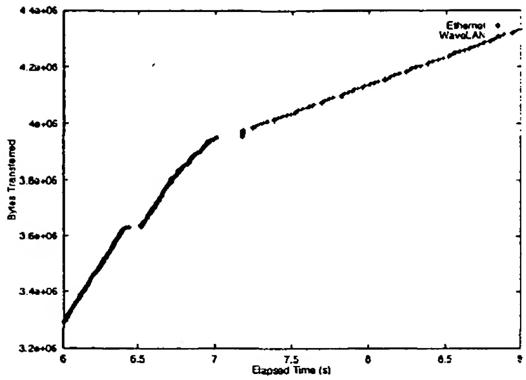

Fig. 7. TCP performance with Mobile-IP on a network with negligible delay

gestion in the system, and enforces congestion control. Figure 7 presents the performance of TCP over Mobile-IP and over MSM-IP. In the case of Mobile-IP, TCP's congestion window shrinks to 1. After that, TCP recovers linearly. In the case of MSM-IP, there is no observable change in TCP's behavior because there is no registration delay incurred. TCP's fast retransmit mechanism significantly improves TCP's performance when used with Mobile-IP, as seen from figure 8. However, it will make no difference under MSM-IP.

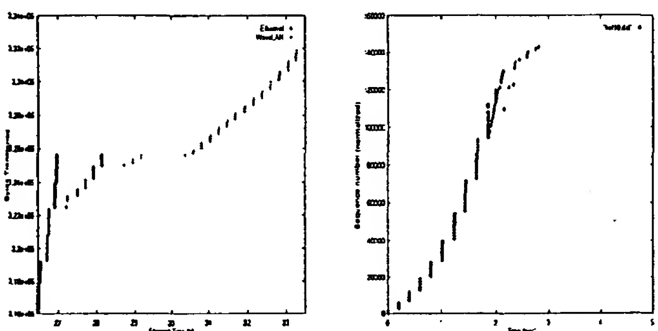

Fig. 8. TCP(without fast retransmit) performance with Mobile-IP(fig. on left) and MSM-IP(right) on a network with 100ms delay

4. RELATED WORK

In this section we compare the MSM-IP approach to other contemporary approaches to providing mobile host support. A common feature of existing approaches for providing network level support for host mobility is the use of a two tier addressing architecture, consisting of a home address and a care-of address. As described in [3], the key functions of any architecture for mobility support include address translation (mapping of the home address to the care-of address), packet forwarding (the tunneling of packets destined to the home address to the location of the care-of address), and location management (discovery and update of the mobile host's location - typically transparent to the correspondent host). Based on these functionalities, [3] compares the following approaches: Columbia [12], Sony [23], Loose Source Routing [17], Mobile IP [20], and IPv6 [18]. Other approaches to support host mobility include Mosquitonet [5] and Daedalus [21].

The approach in Mosquitonet is very similar to the instance of Mobile IP in which a stripped down version of the foreign agent is co-located with the mobile host. Essentially, a mobile host may acquire a care-of address using a protocol such as DHCP, and then encapsulate or decapsulate packets itself, thereby performing the packet forwarding functionality of the foreign agent.

The approach in Daedalus is to multicast packets from the home agent to the cluster of foreign agents in the neighborhood of the mobile host. As the mobile host moves, this cluster changes. This approach is similar to the MSM-IP approach in its goal of using multicasting to reduce handoff latency. However, it still retains the two-level addressing hierarchy of Mobile IP, and thus incurs the same benefits and overheads (e.g. triangular routing) of Mobile IP. In this approach, multicasting is provided only as a convenience for the sole purpose of reducing packet loss upon handoff. In MSM-IP, multicasting is the *only* form of routing to a mobile host. Our whole point is that once an effective IP multicasting infrastructure is in place, it can be used as-is for supporting mobility.

The Columbia approach [12] was among the pioneering efforts to support mobility in the Internet. MSM-IP and the Columbia approach share several common goals and features; in fact, MSM-IP can be viewed as the multicasting analogue of the Columbia approach. In the Columbia approach, mobile hosts belong to a virtual mobile network with a distinct network id. A collection of dedicated mobile support routers (MSRs) are used to provide packet forwarding and location management. MSRs communicate with each other by means of tunnels. Mobile host locations are updated and propagated by means of a distributed directory protocol. The key distinction between the Columbia approach and the MSM-IP approach is the use of multicasting in order to reduce handoff packet loss, the ability to advance register and perform resource reservation, and the use of the existing IP multicasting infrastructure to accomplish host mobility.

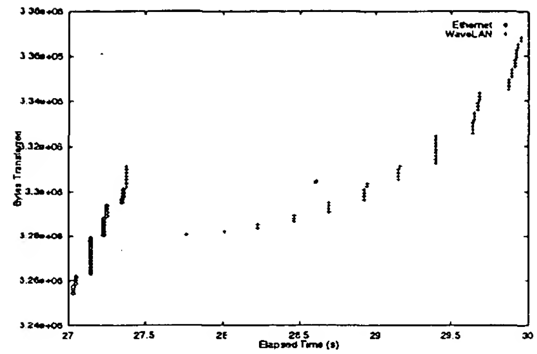

Fig. 9. TCP performance with Mobile-IP on a network with 100ms delay and fast retransmit enabled

5. CONCLUSIONS

In this paper, we have presented the performance of transport protocols over a new architecture for supporting host mobility in the Internet. The mechanism uses the IP multicasting as the sole mechanism for routing packets to the mobile hosts. Our approach, called MSM-IP, assigns a unique location independent (multicast)

address to a mobile host. Consequently, we eliminate the problem of mapping location independent to location dependent addresses for mobile hosts, which is the key focus of almost every contemporary approach for supporting host mobility in the Internet. Since our mobile hosts are identified by multicast addresses, packets from the correspondent host to the mobile host will be tunneled through a sequence of multicast routers in the multicast distribution tree and reach the mobile host, rather than go through a home network as in Mobile IP.

As a result of using multicasting for supporting host mobility, performing advance registration and having packets delivered to the next cell in advance of a handoff is very natural. Likewise, advance resource reservation and the use of RSVP-like mechanisms for resource reservation are also very natural in our approach. IP multicasting seeks to support efficient joins and prunes of receivers in a multicast group. This helps us perform handoffs in an efficient and graceful manner.

We have implemented a testbed and extensively tested its performance for handoffs and interface changes within and across networks, with hot and cold switches, with TCP, UDP and a variety of communication-intensive applications. Our results indicate that MSM-IP performs very well for all the above tests, and is definitely a viable alternative to contemporary approaches for supporting mobility from the perspective of performance.

However, a number of factors currently prevent the deployment of the IP multicasting infrastructure as-is to support host mobility. Scalability, Location Management and security are issues that need to be resolved effectively. A more detailed description of these issues can be found in [16]. Future work will involve a more detailed study of these issues and attempt to provide better and more scalable solutions to the problems that are currently in the way of allowing an MSM-IP architecture to be widely used for supporting host mobility in the Internet.

6. References

[1] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz. "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *Proceedings of ACM SIGCOMM*, Stanford, California, August 1996.

[2] A. Ballardie, P. Francis, and J. Crowcroft. "Core Based Trees," *Proceedings of ACM SIGCOMM*, San Francisco, California, August 1993.

[3] P. Bhagwat, C. Perkins, and Satish K. Tripathi. "Network Layer Mobility : an Architecture and Survey," **IEEE Personal Communication** v. 3 no. 3, June 1996.

[4] R. Caceres and V. Padmanabhan. "Fast and Scalable Handoffs for Wireless Internetworks," *Proceedings of ACM Mobicom*, White Plains, New York, November 1996.

[5] S. Cheshire and M. Baker. "Internet Mobility 4x4," *Proceedings of ACM SIGCOMM*, Stanford, California, August 1996.

[6] Y. Dalal and R. Metcalfe. "Reverse Path Forwarding of Broadcast Packets," *Communications of the ACM*, v. 21, no. 12, December 1978.

[7] S. Deering and J. Hinden. "Internet Protocol, Version 6 (IPv6) Specification," *RFC 1883*, December 1995.

[8] S. Deering, D. Estrin, et al. "Protocol Independent Multicast - Sparse Mode (PIM-SM): Motivation and Architecture," *draft-ietf-idmr-pim-arch-04.ps*, October 1996.

[9] S. Deering and D. Cheriton. "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems*, v.8, no. 2, May 1990.

[10] R. Droms. "Dynamic Host Configuration Protocol," *RFC 1541*, October 1993.

[11] M. Stangel, V. Gupta, and V. Bharghavan. "TCP Connection Management for Lossy Mobile Handoff," *Proceedings of IEEE International Conference on Communications*, June 1998.

[12] J. Ioannidis, D. Duchamp, and G.Q. Maguire Jr.. "IP-based Protocols for Mobile Internetworking," *Proceedings of ACM SIGCOMM*, September 1991.

[13] J.C. Lin and S. Paul. "RMTP: A Reliable Multicast Transport Protocol," *Proceedings of IEEE Infocom*, April 1996.

[14] S. Lu and V. Bharghavan. "Adaptive Resource Management Algorithms in Mobile Computing Environments," *Proceedings of ACM SIGCOMM*, August 1996.

[15] J. Moy. "Multicast Extension to OSPF," *Internet Draft*, September 1992.

[16] J. Mysore and V. Bharghavan. "A New Multicasting-based Arthitecture for Internet Host Mobility," *Proceedings of ACM Mobicom*, September 1997.

[17] C. Perkins and P. Bhagwat. "A Mobile Networking System based on Internet Protocol," *IEEE Personal Communications Magezine*, v.1, no. 1, February 1994.

[18] C. Perkins and D. Johnson. "Mobility Support in IPv6," *draft-ietf-mobileip-ipv6-00.txt*, January 1996.

[19] C. Perkins and D. Johnson. "Mobility Support in IPv6," *Proceedings of ACM Mobicom*, November 1996.

[20] C. Perkins, Editor. "IP Mobility Support," *Network Working Group, RFC 2002*, October 1996.

[21] S. Seshan, H. Balakrishnan and R. H. Katz. "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience," *ACM/Baltzer Journal on Wireless Networks*, 1995.

[22] G. Wright and W. Stevens. "TCP/IP Illustrated, Volume 2," *Addison Weslay*, 1995.

[23] F. Teraoka, Y. Yokote, and M. Tokoro. "A Network Architecture Providing Host Migration Transparency," *Proceedings of ACM SIGCOMM*, 1991.

[24] D. Waitzman, S. Deering, and C. Partridge. "Distance Vector Multicast Routing Protocol," *RFC 1075*, November 1988.

[25] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. "RSVP: A new Resource reSerVation Protocol," *IEEE Network*, September 1993.